Statistical Dependency Parsing of Four Treebanks

Atanas Chanev
University of Trento and ITC-irst Povo-Trento, via Matteo del Ben 5, Rovereto, TN, Italy
E-mail: chanev@form.unitn.it and http://polorovereto.unitn.it/∼chanev

Keywords: human language technologies, dependency parsing, coordination

Received: July 17, 2006

Multilingual dependency parsing is gaining popularity in recent years for several reasons. Dependency structures are more adequate for languages with freer word order than the traditional constituency notion. There is a growing availability of dependency treebanks for new languages. Broad coverage statistical dependency parsers are available and easily portable to new languages. Dependency parsing can provide useful contributions in areas such as information extraction, machine translation and question answering, among others. In addition, syntactic head-dependent pairs are a good interface between the traditional phrase structures and semantic theta roles. In this paper we present the learning curves of a statistical dependency parser for four languages: Arabic, Bulgarian, Italian and Slovene. We discuss issues that mostly concern the employed annotation scheme for each treebank with an emphasis on coordinated structures.

Povzetek: Opisano je večjezično odvisnostno skladenjsko razčlenjevanje štirih jezikov.

1 Introduction

Contrary to a constituency (or phrase structure) grammar, a dependency grammar (e.g. [11]) does not view syntactic structures as nested sets of constituents but as a set of binary head-dependent relations. In most dependency grammar formalisms there are several restrictions for the dependency relations: They should build up a connected acyclic graph; For each dependent, there should be only one head; There should be a single word in the sentence without a head – the root word. A syntactic label, such as subject, object etc. is usually associated with each relation in the graph.

Projectivity is another issue that is often considered as a constraint to dependency graphs. A simple non-formal definition for projectivity of a connected dependency graph is: if one connects the root word of a sentence with an artificial root placed before the first word, there should not be crossing dependency arcs. While most of the dependency parsers can parse only projective structures, the need for non-projective relations is recognised in nearly all dependency treebank annotation schemes.

State-of-the art statistical dependency parsers have been evaluated on 13 different treebanks (for 13 different languages) at the CoNLL-X shared task on statistical dependency parsing [2][1]. While the treebanks had been parsed with many parsers, all the parsers had been implementations of a limited number of parsing models.

This paper gives the learning curves for four languages (Arabic, Bulgarian, Italian and Slovene) of one of the parsers tested at the CoNLL-X shared task – Maltparser [12]. The parser has a high attachment score (accuracy), and it is robust. The treebanks for Arabic, Bulgarian, Italian and Slovene had been annotated by different research groups, using four different annotation schemes.

The paper is structured as follows: Section 2 explains our motivation to choose Maltparser among the CoNLL-X shared task parsers for our experiments. Then, in Section 3 we briefly describe the properties of each treebank that we give learning curves of. We give a short description of Maltparser and the parsing feature model that we used in our experiments in Section 4. The learning curves are given in tabulated form and discussed in Section 7. We conclude in Section 6.

2 Motivation for our choice of a parser

We chose Maltparser [12] from the pool of CoNLL-X shared task parsers because of its high (second best) overall accuracy in the CoNLL-X shared task. Furthermore, it has a number of desired properties which are consistent with our long term goal – to use a broad coverage automatic parser as a model of the human parsing mechanism. Such properties include the ability different types of information to be used in feature models.

Maltparser employs one of the two most commonly used parsing models at the shared task. The dependency graph is built using a stack for storing the words of the sentence and four actions: shift, reduce, left-arc and right-arc. Projectivity of the treebank to be learned and parsed can be ‘enforced’ using pre and post processing graph transformations. However, we did not take benefit from that option since we do not believe that such transformations are plausible in the human parsing sense.

[1] http://nextens.uvt.nl/∼conll/

3 Treebanks

We used four treebanks in our experiments: The Prague Arabic Dependency Treebank (PADT) [7], the BulTreeBank (BTB) [13], the Turin University Treebank (TUT) [1] and the Slovene Dependency Treebank (SDT) [6]. PADT, TUT and SDT are original dependency treebanks while BTB was converted from Head-driven Phrase Structure Grammar (HPSG) format to dependency graphs in [3].

3.1 The Prague Arabic Dependency Treebank

We used the CoNLL-X shared task version of the PADT[2] which differs slightly from the original treebank. It is separated in training (1,460 sentences; 54,379 tokens) and test (146 sentences; 5,373 tokens) set. The number of part-of-speech tags and the number of dependency tags are respectively 21 and 27. The average number of tokens per sentence is 37.2. The PADT annotation scheme is closely related to the one of the Prague Dependency Treebank (PDT) [8].

One of the idiosyncrasies of the PDT annotation is the treatment of coordinated structures. In PDT-related annotation schemes the coordinating conjunction (or punctuation) is the head of the coordinated words.

3.2 The BulTreeBank

BulTreeBank is an HPSG-based treebank but head-dependent relations between words are not stated explicitly. It has been converted to dependency graph representations in [3]. We use the CoNLL-X shared task dependency version of the BTB for our results to be comparable to those from the CoNLL-X shared task.

The BulTreeBank is separated in training (10,911 sentences; 159,395 tokens) and test (2,310 sentences; 36,756 tokens) set. The average number of words per sentence is 14.8. The number of part-of-speech labels is 570[3] and the number of dependency labels is 20.

Coordinated structures are annotated differently from those in the PADT. In the BTB encoding the first coordinated word is annotated as the head of the coordinating conjunction (or punctuation) and as the head of the second coordinated word.

3.3 The Turin University Treebank

The TUT was not included in the CoNLL-X shared task mainly because of its limited size – 1,500 sentences (44,616 tokens). The average number of tokens per sentence is 27.7. Although the treebank is small and n-fold cross-validation is usually used in such cases, here we report results on a test set of 150 sentences (4,172 tokens) and a training set of 1,350 sentences (37,444 tokens) in order the TUT experiments not to differ from the experiments on the other treebanks in this study.

We used a version of the TUT with removed traces and reduced tag sets [4] (90 part-of-speech tags and 18 dependency tags). Italian dependency tags are semantically 'deeper' than those from the other treebanks in this study. Coordination is annotated with the coordinating conjunction (or punctuation) being the head of the second coordinated word and a dependent of the first coordinated word.

3.4 The Slovene Dependency Treebank

SDT has an annotation scheme which is similar to those of the PDT and PADT. We used the CoNLL-X version of the treebank for our results to be comparable with those from the shared task. The data is divided in a training set (1,534 sentences, 28,750 words) and a test set (402 sentences, 6,390 words). The average number of tokens per sentence is 18.2. The number of the part-of-speech tags used in the annotation of SDT is 30. The number of dependency labels is 26. Like in PADT, coordinated structures are treated with the coordinating conjunction (or punctuation) being the head of the coordinated words.

4 The parser

We used version 0.4 of Maltparser[4]. Maltparser does not use an explicit probabilistic grammar but implements a data-driven parsing approach. What is learned is the actions that the parser must take in order to build the dependency graph of the sentence. We used the Support Vector Machines (SVM) learner [5] which is included in Maltparser 0.4. PoS tags, words as well as dependency labels which have already been assigned by the parser on the run can be used in feature models for learning.

We employed a common feature model (m7) which consists of six part-of-speech features, four dependency features and four lexical features. More information about the parser and feature models can be found on the Maltparser web page. The Maltparser team reported the second best result at the CoNLL-X shared task [12] (the difference from the best result is not statistically significant).

5 Results

In this section we list related work, describe preliminary settings, present in tabular form and discuss the learning curves for Arabic, Bulgarian, Italian and Slovene. The measure that we use for evaluation is labelled attachment score (labelled accuracy) measured excluding punctuation. We also report unlabelled attachment score (unlabelled accuracy). For a definition of these measures, the reader is referred to [10].

[2] PADT is distributed by the Linguistic Data Consortium: http://www.ldc.upenn.edu/

[3] We used the original BTB part-of-speech tags.

[4] http://w3.msi.vxu.se/∼nivre/research/MaltParser.html

5.1 Previous studies

In this section we give only dependency (and not constituency) parsing results because they are immediately relevant to the study.

5.1.1 Arabic

The PADT has been learned and parsed by various teams at the CoNLL-X shared task on dependency parsing. Results vary from 50.7% to 66.9% labelled accuracy [2].

5.1.2 Bulgarian

A dependency version of the BulTreeBank has also been used at the CoNLL-X shared task. Labelled accuracy is within the range 67.6% – 87.6%. Labelled accuracy of 79.5% was reported for another conversion of the original HPSG-based BulTreeBank but those results did not differ significantly from the results reported on the CoNLL-X conversion using the same parser and feature model (79.2%) [3].

5.1.3 Italian

We will compare the learning curves for Italian with [4] where a previous version of the Maltparser was used together with another learner. The reported accuracy is 81.8%. A rule-based dependency parser for Italian is described in [9]. Even though its evaluation is only partial, its accuracy is comparable to the one reported in [4].

5.1.4 Slovene

Slovene, like Arabic and Bulgarian, was one of the languages for the CoNLL-X shared task. Results for Slovene varied from 50.7% to 73.4% labelled accuracy [2].

5.2 Settings

All the experiments were performed on training and test sets with gold standard PoS tags. The same feature model and the same learning and parsing settings were used in all the tests with the exception of an option that we used only for the Arabic and Slovene treebanks where graphs may be interpreted as having multiple roots.

The BulTreeBank learning curve is set for training sets that start from 1,000 sentences and increase up to the full size of the treebank, where at each step the size of the training set is increased by 1,000 sentences. The learning curves for the other languages start from a training set of 600 sentences and the sizes continue to grow up to the full number of sentences of the treebanks with increase of 200 sentences at each step.

Two additional learning curves are included for Arabic and Slovene after a simple graph transformation on the coordinated structures was applied on the training sets for these languages. Parsing output was then converted back to the original coordination encoding and evaluated on the gold standard PADT and SDT.

A description of the coordination transformation procedure follows:

Coordinated structures are identified by the dependency label of the coordinating conjunction (or punctuation) which, according to the PDT annotation scheme, is the head of the coordinated words. If there are two words with the same dependency labels among the dependents, one of them being before the head and the other – after the head, they are recognised as coordinated. Then the first coordinated word takes the head word of the coordinating conjunction (punctuation) and the coordinating conjunction or punctuation is made to point to the first coordinated word.

The inverted transformation is performed in a similar way. After the coordinated structure is identified, the head of the first coordinated word is transferred to be the head of the coordinating conjunction (or punctuation) and the first coordinated word is made dependent on the coordinating conjunction (or punctuation). Note that the back transformation can be accurate only for properly parsed coordinated structures.

5.3 Learning curves

The learning curves are given in tabular form in Tables from 1. to 4. The first column shows the training set size in sentences. AS_L and AS_U stay respectively for labelled and unlabelled attachment score.

Size	AS_L	(c.t.)	AS_U	(c.t.)
600	61.1%	61.9%	73.0%	74.0%
800	62.4%	64.2%	74.2%	76.0%
1,000	63.9%	65.2%	75.1%	76.6%
1,200	65.2%	67.1%	75.7%	78.1%
1,400	65.6%	67.4%	76.2%	78.2%
1,460	66.0%	67.4%	76.3%	78.0%

Table 1: Learning curves for the PADT (Arabic). c.t. = Coordination transformation applied.

For training data of 1,000 sentences labelled accuracies for Bulgarian, Slovene and Arabic are similar. Labelled accuracy for Italian is the highest for this size of training data. If the comparison is done using the unlabelled accuracy measure, the per cent for Bulgarian is lower than those for Arabic and Slovene due to the bigger difference between labelled and unlabelled accuracy for PADT and SDT.

There are a number of reasons for the differences in accuracy for the different treebanks, from numbers of tokens per sentence for each treebank to sizes of the tag sets and idiosyncrasies of the annotation schemes. For example, the small number of part-of-speech tags for the Arabic and Slovene treebanks might have been the reason for the lower accuracy, in comparison with the bigger number of

Size:	AS$_L$	AS$_U$
1,000	64.8%	71.6%
2,000	69.4%	75.8%
3,000	75.6%	81.3%
4,000	77.5%	83.1%
5,000	78.4%	83.8%
6,000	79.8%	85.0%
7,000	80.0%	85.2%
8,000	80.4%	85.6%
9,000	80.9%	86.0%
10,000	81.5%	86.5%
10,911	81.8%	86.8%

Table 2: Learning curves for the BTB (Bulgarian).

Size:	AS$_L$	AS$_U$
600	80.9%	86.5%
800	82.3%	87.7%
1,000	82.8%	88.2%
1,200	83.0%	88.3%
1,350	83.7%	88.6%

Table 3: Learning curves for the TUT (Italian).

PoS tags for the Italian treebank, given that the number of dependency tags is similar in all the three treebanks. In fact, this is not the case. We did an additional experiment on the Italian data. We used a PoS set of only 17 coarse grained tags and labelled accuracy was still above 81% for the biggest training set.

The Arabic and Slovene data sets had their transformed versions learned and parsed better than the original ones. The difference in labelled accuracy is over 1% for nearly all the sets. The biggest training set gives worse results than the second biggest for the transformed Slovene training data. This is due to loss of accuracy in the inverted transformation.

The number of non-projective trees in the treebanks have influenced parsing accuracy since the parser cannot parse non-projective arcs. Non-projective trees are 175 (10.9%) in PADT, 962 (7.3%) in BTB, 91 (6.1%) in TUT[5] and 1,289 (66.6%) in SDT. The number of sentences with the hard-to-parse coordinated structures in the PADT and SDT are respectively 1,041 (64.8%) and 989 (51.1%).

The results for Arabic reported in this paper are slightly higher (0.5%) than the best results reported at the CoNLL-X shared task even though a more sophisticated feature model for the Maltparser was used there.

Results for Bulgarian are lower, if compared to the Maltparser results obtained at the CoNLL-X shared task where it employed a better feature model. The accuracy that we report here is higher than the one reported in [3] because they used an option of the SVM learner which split the data on smaller parts for faster learning with the cost of decrease in performance.

Size:	AS$_L$	(c.t.)	AS$_U$	(c.t.)
600	62.3%	63.7%	73.8%	74.1%
800	64.0%	65.6%	74.8%	75.7%
1,000	64.6%	66.4%	75.2%	75.9%
1,200	65.6%	67.0%	75.9%	76.4%
1,400	66.8%	68.3%	77.0%	77.5%
1,534	67.1%	68.2%	77.4%	77.6%

Table 4: Learning curves for the SDT (Slovene). c.t. = Coordination transformation applied.

Compared to the other treebanks the parser learned the TUT very well with a limited amount of training data. The reason for the high accuracies is likely to be the treebank annotation scheme. It is different from those of the other treebanks in its 'deeper' syntactic dependency relations. The distance between the dependents and their heads is usually short which facilitates processing. The number of sentences in TUT which have non-projective graphs is very small – only 91. That may have contributed to the high parsing accuracy. Compared to previous studies we report higher accuracy (nearly 2% increase).

Our results for Slovene somehow lag behind the results for that language which were obtained using Maltparser at the CoNLL-X shared task. The reasons are the use of a simpler feature model for the parser and the big number of non-projective arcs in the Slovene treebank which we did not pre/post processed.

Results are on the average 1% higher than those for the PADT. Possibly this difference can be explained with the very small number of tokens per sentence for the SDT – only 18.2, compared to 37.2 for the Arabic treebank. As in the case with PADT, coordination transformations increased parsing accuracy.

6 Conclusion and future work

We presented the learning curves for four different treebanks using the same feature model for learning a statistical dependency parser. We showed that often parsing results differ significantly for different languages and the reasons can be various properties of the concrete treebank. We performed treebank transformations for Arabic and Slovene to report parsing accuracy for Arabic that is slightly higher than the best results reported at the CoNLL-X shared task.

Future work includes investigation of various treebanks to find out which annotation scheme keeps parsing accuracy high for a vast majority of languages. In addition we believe that adding different kind of information as features in the learning model can lead to broad coverage models of the human sentence parsing mechanism whose implementations must be good multilingual NLP parsers.

[5] Originally the TUT does not have non-projective sentences but after traces were removed in [4] non-projective arcs were introduced.

References

[1] C. Bosco (2004) *A grammatical relation system for treebank annotation*, PhD thesis, University of Turin.

[2] S. Buchholz and E. Marsi (2006) CoNLL-X shared task on multilingual dependency parsing, *Proc. of CoNLL-X*, Omnipress, New York, pp. 149–164.

[3] A. Chanev, K. Simov, P. Osenova, S. Marinov (2006) Dependency Conversion and parsing of the BulTreeBank, *Proc. of the LREC workshop Merging and Layering Linguistic Information*, Genoa, pp. 17–24.

[4] A. Chanev (2006) Portability of parsing algorithms – an application for Italian, *Proc. of the Treebanks and Linguistic Theories workshop*, Barcelona, pp. 29–40.

[5] C.-C. Chang and C.-J. Lin (2005) *LIBSVM: A library for support vector machines*, www: csie.ntu.edu.tw/∼cjlin/papers/libsvm.pdf

[6] S. Džeroski, T. Erjavec, N. Ledinek, P. Pajas, Z. Žabokrtsky and A. Žele (2006) Towards a Slovene dependency treebank, *Proc. of the Fifth International Conference on Language Resources and Evaluation*, ELRA, Genoa.

[7] J. Hajič, O. Smrž, P. Zemánek, J. Šnaidauf and E. Beška (2004) Prague Arabic Dependency Treebank: Development in data and tools, *Proc. of the NEMLAR International Conference on Arabic Language Resources and Tools*, Cairo, pp. 110–117.

[8] J. Hajič (1998) Building a syntactically annotated corpus: The Prague Dependency Treebank, *Issues of Valency and Meaning*, Karolinum, pp. 12–19.

[9] L. Lesmo, V. Lombardo and C. Bosco (2002) Treebank development: The TUT approach, *R. Sangal and S.M. Bendre, ed. Recent Advances in Natural Language Processing*, Vikas Publ. House, New Delhi, pp. 61–70.

[10] D. Lin (1998) A dependency-based method for evaluating broad-coverage parsers, *Natural Language Engineering 4(2)*, Cambridge University Press, pp. 97–114.

[11] I. Mel'čuk (1988) *Dependency syntax: Theory and practice*, State University of New York Press.

[12] J. Nivre, J. Hall, J. Nilsson, G. Eryiğit and S. Marinov (2006) Labeled pseudo-projective dependency parsing with Support Vector Machines, *Proc. of CoNLL-X*, Omnipress, New York, pp. 221–225.

[13] K. Simov, P. Osenova, A. Simov and M. Kouylekov (2005) Design and implementation of the Bulgarian HPSG-based treebank, *Journal of Research on Language and Computation – Special Issue*, Kluwer Academic Publishers, pp. 495–522.